

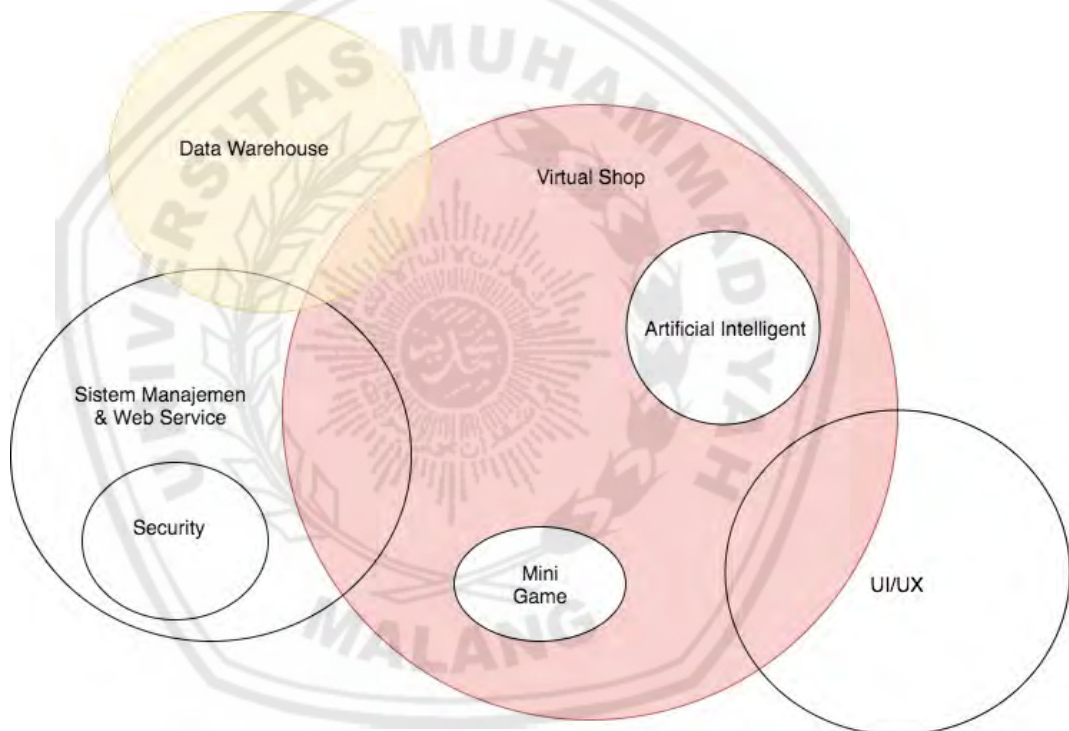
## BAB III

### PERANCANGAN

Pada bab ini membahas perancangan pada data warehouse. Perancangan *Data Warehouse* meliputi perancangan *ER Diagram* untuk *OLAP* menggunakan *Metode From ER Models to Dimensional Models* dari Moody.

#### 3.1 Metode Penelitian

Berikut merupakan gambaran secara umum sistem *Virtual Shop* yang dimana terdiri dari 7 bagian. Pada penelitian ini, bagian yang di ambil adalah *Data Warehouse*



**Gambar 3.1** Gambaran Umum Sistem Virtual Shop

Gambar 3.1 dapat dilihat tentang gambaran keseluruhan sistem Virtual Shop, penulis akan melakukan penelitian mengenai pembangunan *Data Warehouse* untuk *Virtual Shop*. Dalam Pembangunan *Data Warehouse* dilakukan dalam beberapa tahapan perancangan sesuai dengan metodologi perancangan data warehouse yaitu *From ER Models to Dimensional Models* dari Moody.

### 3.2 ER Diagram Database

Pada sub bab ini akan dijelaskan rancangan yang berkaitan dengan Diagram OLTP pada *Virtual Shop*. *ER Diagram* meliputi *ER Diagram* pada *OLTP Virtual Shop*.

#### 3.2.1 ER Diagram OLTP Virtual Shop

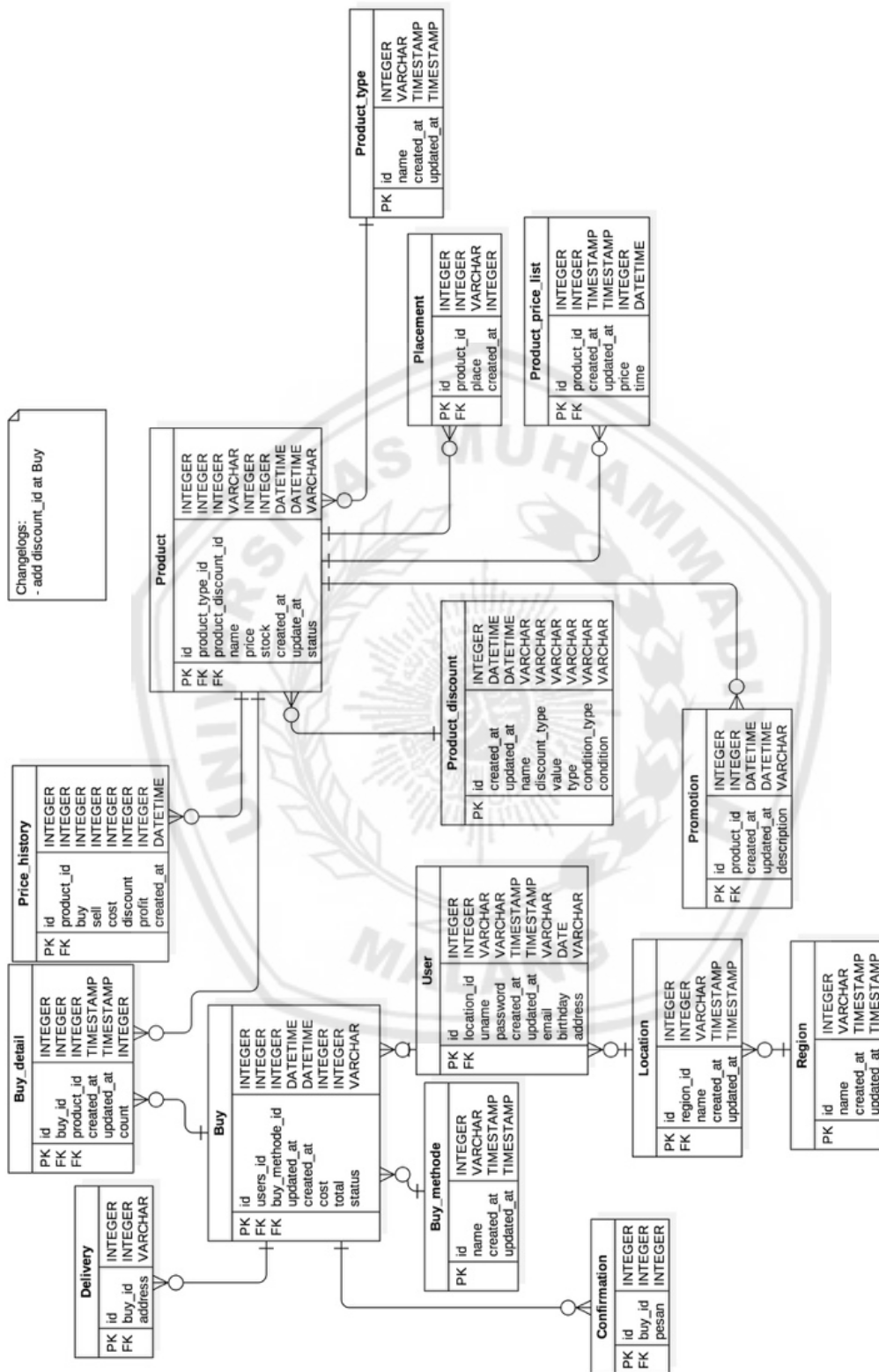
ER Diagram OLTP pada Virtual Shop merupakan rancangan yang digunakan pada sistem manajemen Virtual Shop. Rancangan ini merujuk pada penelitian yang dilakukan oleh Gusti Alfian M.P yang berjudul *Implementasi Rest dan Sistem Manajemen Pada Virtual Shop*. Pada *OLTP* tersebut memiliki total tabel sebanyak 16 tabel yaitu

**Tabel 3.1** Deskripsi Tabel *OLTP Virtual Shop*

| No. | Nama Tabel   | Deskripsi  |
|-----|--------------|--|
| 1   | Buy          | Tabel ini memuat data yang berkaitan dengan transaksi pembelian                        |
| 2.  | Buy_detail   | Tabel ini berisi data yang berkaitan dengan detail pembelian yang dilakukan            |
| 3.  | Buy_methode  | Tabel ini  |
| 4.  | Confirmation | Tabel ini berisi data pembelian yang memerlukan konfirmasi                             |
| 5.  | Delivery     | Tabel ini berisi data yang berkaitan dengan progres pengiriman dari barang yang dibeli |
| 6.  | Location     | Tabel ini berisi data yang berkaitan dengan lokasi dari user                           |
| 7.  | Placement    | Tabel ini berisi data yang berkaitan dengan penempatan barang yang ada                 |

|     |                    |   |
|-----|--------------------|---|
| 8.  | Price_history      | Tabel ini berisi data yang berkaitan dengan perubahan harga pada barang yang dijual |
| 9.  | Product            | Tabel ini berisi data yang berkaitan dengan deskripsi umum produk yang dijual       |
| 10  | Product_discount   | Tabel ini berisi data yang berkaitan dengan   |
| 11. | Product_location   | Tabel ini berisi data yang berkaitan dengan lokasi produk berada                    |
| 12. | Product_price_list | Tabel ini memuat data berkaitan dengan list harga pada produk yang dijual           |
| 13. | Product_type       | Tabel ini memuat data yang berkaitan dengan tipe produk                             |
| 14. | Promotion          | Tabel ini memuat data yang berkaitan dengan promosi yang berlaku pada produk        |
| 15. | Region             | Tabel ini memuat data regional dari user  |
| 16. | User               | Tabel ini memuat data deskripsi user  |

Pada rancangan *OLTP* tersebut akan di kondisikan sesuai dengan paradigma skema *data warehouse* yang akan digunakan yaitu *Snowflake Schema* dengan menerapkan metodologi perancangan *data warehouse From ER Models to Dimensional Models* dari Moody

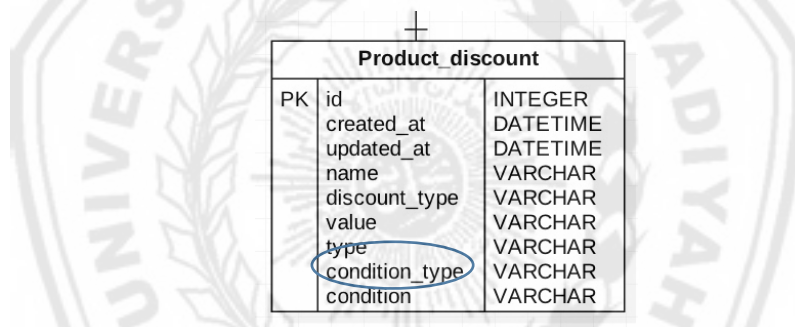


Gambar 3.2 ER Diagram OLTP pada Virtual Shop

Pada Gambar 3.6 dapat dilihat bahwa pada tabel *Product\_discount* terdapatnya kolom *condition\_type* dan *condition*. Kondisi tersebut dapat menyebabkan terjadinya perulangan data atau bisa disebut dengan *redundansi data*. Pada penelitian ini menerapkan paradigma *snowflake schema* sehingga tabel *OLTP* tersebut perlu dilakukan pengkondisian dengan melakukan *normalisasi*. *Normalisasi* yang dapat dilakukan adalah melakukan pemecahan pada tabel tersebut sehingga terbentuk tabel baru berdasarkan kolom tersebut.

### 3.2.1.1 Pembentukan Tabel *Condition\_type*

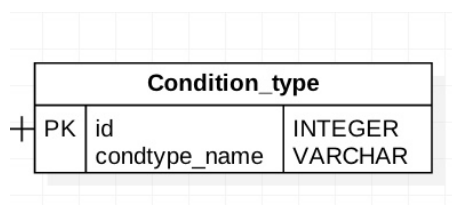
Pembentukan Tabel *Condition\_type* merupakan tahap awal yang dilakukan pada pemecahan tabel *Product\_discount*. Hal tersebut terjadi karena tabel *Condition\_type* merupakan tabel *atomik* atau tabel yang tidak dapat dilakukan pemecahan lagi.



| Product_discount |                |          |
|------------------|----------------|----------|
| PK               | id             | INTEGER  |
|                  | created_at     | DATETIME |
|                  | updated_at     | DATETIME |
|                  | name           | VARCHAR  |
|                  | discount_type  | VARCHAR  |
|                  | value          | VARCHAR  |
|                  | type           | VARCHAR  |
|                  | condition_type | VARCHAR  |
|                  | condition      | VARCHAR  |

**Gambar 3.3** Kolom pada Tabel *Product\_discount*

Kolom tersebut akan dipisahkan kedalam tabel sendiri dengan nama *Condition\_type*.



| Condition_type |               |         |
|----------------|---------------|---------|
| PK             | id            | INTEGER |
|                | condtype_name | VARCHAR |

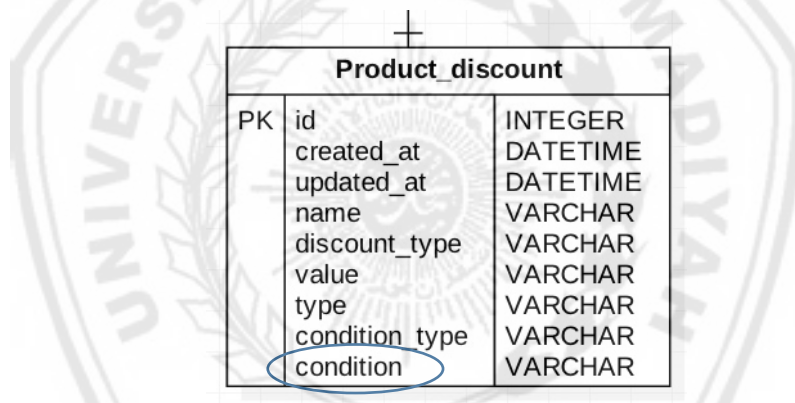
**Gambar 3.4** Tabel *Condition\_type* hasil pemecahan pada Tabel *Product\_discount*

**Tabel 3.2** Kolom pada tabel Condition\_type

| No. | Nama Kolom    | Keterangan                                   |
|-----|---------------|--|
| 1.  | id            | Untuk menyimpan id pada tabel Condition_type |
| 2.  | condtype_name | Untuk menyimpan nama dari tipe kondisi       |

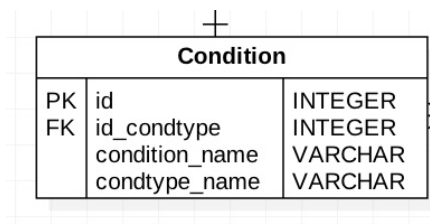
### 3.2.1.2 Pembentukan Tabel Condition

Tabel *Condition* merupakan tabel yang menampung data berkaitan dengan kolom *Condition* pada tabel *Product\_discount*. Tabel ini terbentuk dari teknik pemecahan yang dilakukan pada tabel *Product\_discount* yaitu pada kolom *Condition*.



**Gambar 3.5** Kolom *Condition* pada tabel *Product\_discount*

Tabel *Product\_discount* dilakukan pemecahan menjadi tabel *Condition*. Tabel tersebut berfungsi untuk menampung data dari kolom *condition* pada tabel *Product\_discount*. Hasil pemecahan dapat dilihat pada gambar 3.6

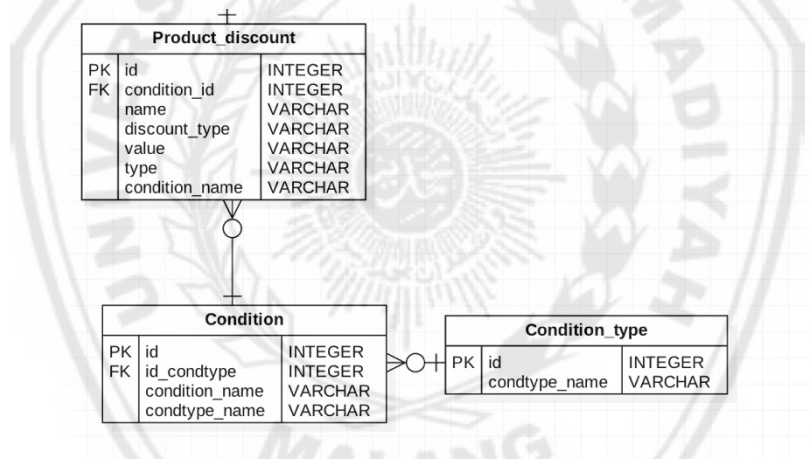


**Gambar 3.6** Tabel *Condition*

**Tabel 3.3** Penjelasan Kolom pada Tabel *Condition*

| NO. | Nama Kolom     | Keterangan  |
|-----|----------------|---|
| 1.  | id             | Kolom untuk menyimpan id / primary key dari tabel Condition |
| 2.  | id_condtype    | Kolom untuk menyimpan id dari tabel Condition_type          |
| 3.  | condition_name | Kolom untuk menyimpan data nama kondisi                     |
| 4.  | condtype_name  | Kolom untuk menyimpan data nama tipe kondisi                |

Hasil pembentukan kedua tabel tersebut menghasilkan susunan yang baru seperti gambar sebagai berikut



**Gambar 3.7** Rancangan setelah dilakukan pembentukan tabel *Condition\_type* dan *Condition*

### 3.3 Metode From ER Model to Dimensional Model

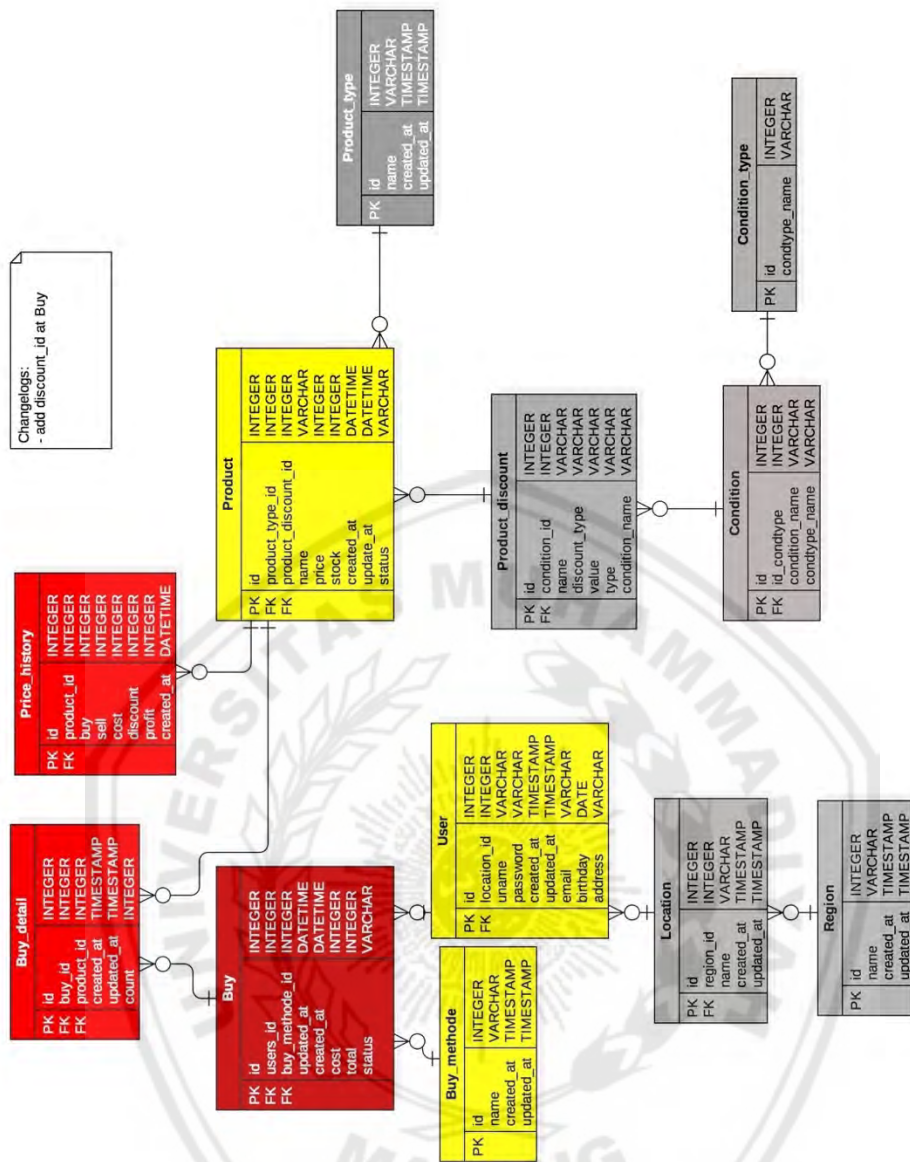
Pada metode ini terdapat empat langkah utama dalam melakukan perancangan *data warehouse*(moody). Pada sub bab berikut ini akan dijelaskan masing-masing tahapan tersebut

### 3.3.1 Klasifikasi Entitas (*Classify Entities*)

Pada bagian ini akan dilakukan pembagian entitas sesuai dengan rancangan *OLTP* pada *Virtual Shop*. Pembagian entitas ada tiga jenis antara lain, *Entitas Transaksi*, *Entitas Komponen*, dan *Entitas Klasifikasi*. Gambar 3.3 adalah gambaran pembagian entitas berdasarkan diagram *OLTP Virtual Shop*.







Entitas Transaksi

Entitas Komponen

Entitas Klasifikasi

Gambar 3.8 Pembagian Entitas pada diagram OLTP Virtual Shop

### 3.3.1.1 Entitas Transaksi

Tabel-tabel yang termasuk pada *Entitas Transaksi* adalah tabel *Buy*, Tabel *Buy\_detail*, dan Tabel *Price\_history*. Tabel *Buy* dapat dikatakan sebagai golongan *Entitas Transaksi* dikarenakan tabel tersebut memuat data perhitungan angka total pembelian. Data tersebut termuat pada kolom *total*.

| Buy |                |          |
|-----|----------------|----------|
| PK  | id             | INTEGER  |
| FK  | users_id       | INTEGER  |
| FK  | buy_methode_id | INTEGER  |
|     | updated_at     | DATETIME |
|     | created_at     | DATETIME |
|     | cost           | INTEGER  |
|     | total          | INTEGER  |
|     | status         | VARCHAR  |

**Gambar 3.9** Kolom *total* pada tabel *Buy* pada OLTP Virtual Shop

Tabel *Buy\_detail* dapat dikatakan sebagai golongan *Entitas Transaksi* dikarenakan tabel tersebut memuat data perhitungan *total* produk yang dibeli. Data perhitungan tersebut termuat pada kolom *count*.

| Buy_detail |            |           |
|------------|------------|-----------|
| PK         | id         | INTEGER   |
| FK         | buy_id     | INTEGER   |
| FK         | product_id | INTEGER   |
|            | created_at | TIMESTAMP |
|            | updated_at | TIMESTAMP |
|            | count      | INTEGER   |

**Gambar 3.10** Kolom *count* pada tabel *Buy\_detail*

Tabel *Price\_history* dapat dikatakan sebagai golongan *Entitas Transaksi* dikarenakan tabel tersebut memuat data perhitungan perubahan harga yang terjadi pada produk yang dijual. Data perhitungan perubahan harga tersebut tersimpan pada kolom *buy*, *sell*, dan *profit*.

| Price_history |            |          |
|---------------|------------|----------|
| PK            | id         | INTEGER  |
| FK            | product_id | INTEGER  |
|               | buy        | INTEGER  |
|               | sell       | INTEGER  |
|               | cost       | INTEGER  |
|               | discount   | INTEGER  |
|               | profit     | INTEGER  |
|               | created_at | DATETIME |

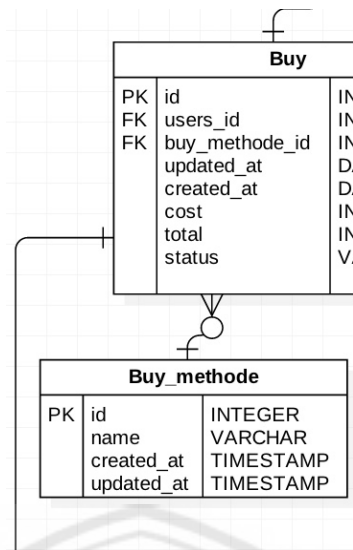
**Gambar 3.11** Kolom *buy* & *sell* pada tabel *price\_history*

### 3.3.1.2 Entitas Komponen

Tabel-tabel yang termasuk pada *entitas komponen* adalah Tabel *Buy\_methode*, Tabel *User*, dan Tabel *Product*. Alasan tabel tersebut digolongkan sebagai Entitas Komponen adalah tabel tersebut terhubung langsung dengan tabel-tabel pada *Entitas Transaksi* (Moody). Berikut adalah contoh yang merepresentasi penggolongan *Entitas Komponen*

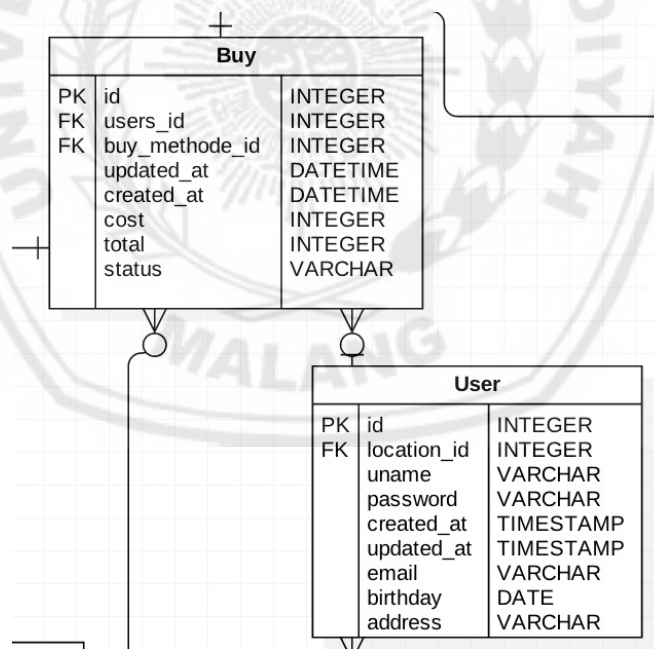
- *Buy\_methode* = Metode apa yang digunakan untuk pembelian
- *User* = Siapa yang melakukan pembelian
- *Product* = Produk apa yang dijual

Tabel *Buy\_methode* dapat digolongkan sebagai *Entitas Komponen* dikarenakan pada tabel ini terhubung langsung secara *one-to-Many* terhadap tabel *Buy* yang termasuk golongan *Entitas Transaksi*.



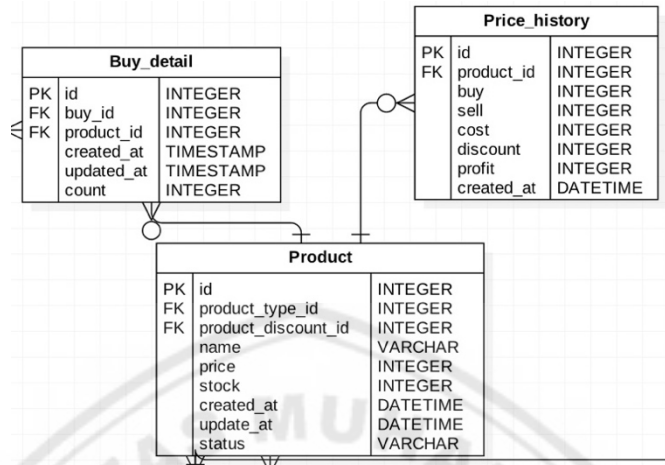
**Gambar 3.12** Relasi tabel *Buy\_Methode* terhadap Tabel *Buy*

Tabel *User* dapat digolongkan sebagai *Entitas Komponen* dikarenakan pada tabel ini terhubung langsung secara *one-to-many* terhadap Tabel *Buy* yang termasuk pada golongan *Entitas Transaksi*.



**Gambar 3.13** Relasi Tabel *User* terhadap Tabel *Buy*

Tabel *Product* dapat digolongkan sebagai *Entitas Komponen* dikarenakan pada tabel ini terhubung langsung secara *one-to-many* terhadap Tabel *Buy\_detail* dan Tabel *Price\_history* yang termasuk pada golongan *Entitas Transaksi*.



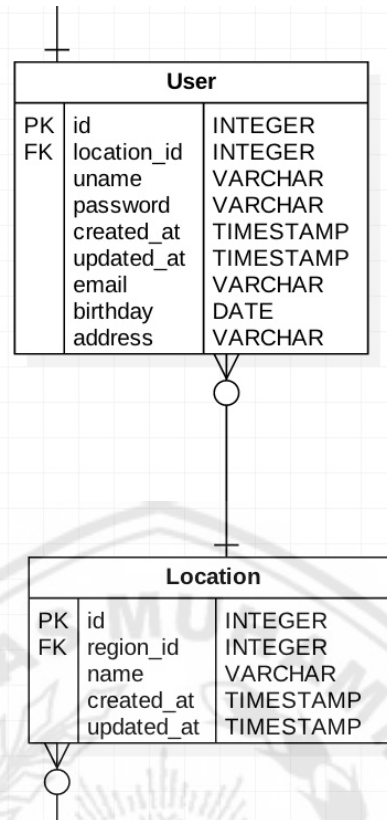
**Gambar 3.14** Relasi Tabel *Product* terhadap Tabel *Price\_history* dan Tabel *Buy\_detail*

### 3.3.1.3 Entitas Klasifikasi

Tabel-tabel yang termasuk pada *Entitas Klasifikasi* adalah Tabel *Location*, Tabel *Region*, Tabel *Product\_discount*, dan Tabel *Product\_type*. Alasan yang mendasari tabel tersebut dapat digolongkan sebagai *Entitas Klasifikasi* adalah tabel tersebut terhubung langsung dengan tabel-tabel pada *Entitas Komponen* dengan relasi *One-to-Many* (Moody).

#### Tabel Location

Tabel *Location* digolongkan pada *Entitas Klasifikasi* dikarenakan terhubung langsung secara *one-to-many* terhadap salah satu tabel pada *Entitas Komponen* yaitu tabel *User*. Hal tersebut dapat dilihat pada gambar 3.15.

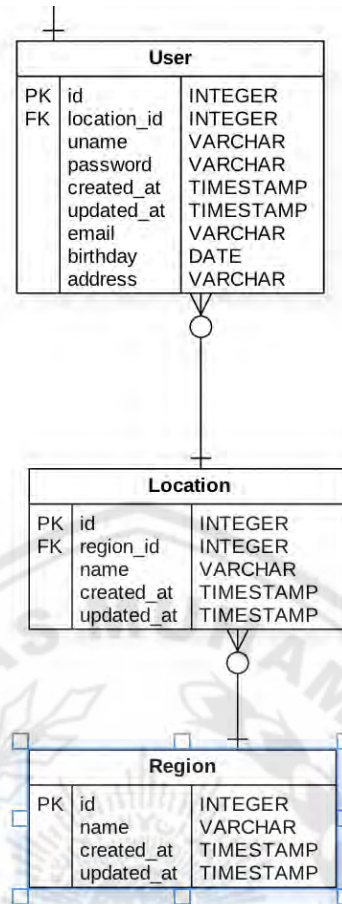


**Gambar 3.15** Relasi Tabel *Location* terhadap Tabel *User*

### Tabel Region

Tabel *Region* digolongkan pada *Entitas Klasifikasi* dikarenakan terhubung tidak langsung terhadap tabel *User*, akan tetapi terhubung langsung secara *one-to-many* terhadap tabel *Location* yang merupakan golongan tabel pada *Entitas Klasifikasi*. Hal ini dapat dijelaskan bahwa tabel yang memiliki relasi langsung terhadap tabel entitas Klasifikasi yang lain dapat juga di golongankan pada Entitas Klasifikasi.

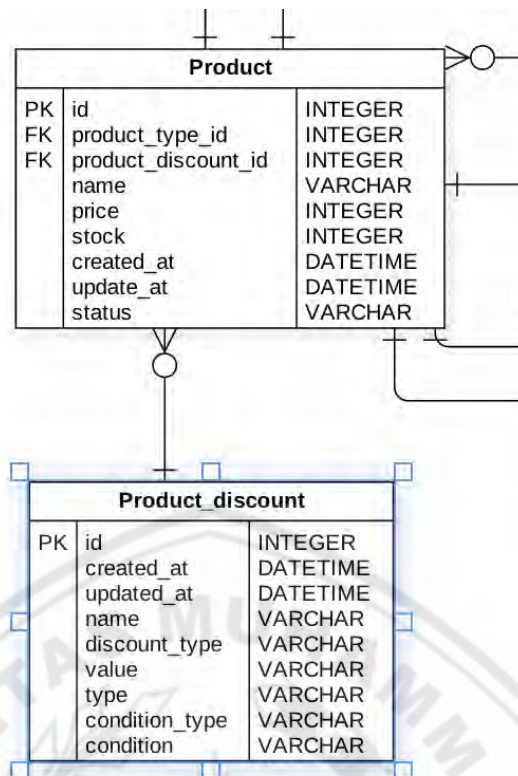
Tabel *Region* memiliki hubungan terhadap tabel *User* melalui tabel *Location* yang dapat dilihat pada gambar 3.16.



**Gambar 3.16** Relasi Tabel Region terhadap tabel Location dan tabel User

#### **Tabel Product\_discount**

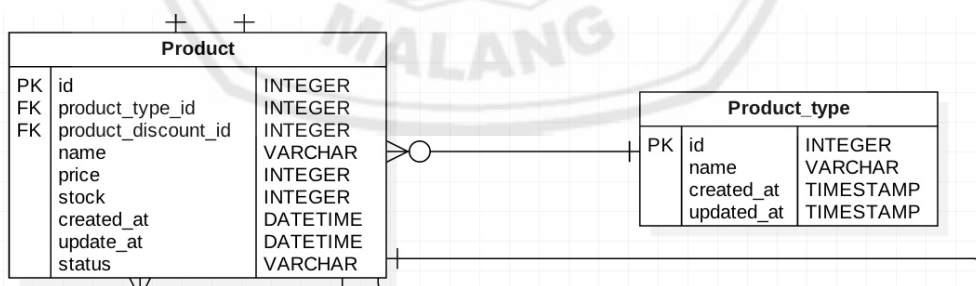
Tabel *Product\_discount* digolongkan pada *Entitas Klasifikasi* dikarenakan terhubung langsung secara *one-to-many* terhadap salah satu tabel pada *Entitas Komponen* yaitu tabel *Product*. Hal tersebut dapat dilihat melalui gambar 3.17



**Gambar 3.17** Relasi tabel *Product\_discount* terhadap tabel *Product*

### Tabel *Product\_type*

Tabel *Product\_type* digolongkan pada Entitas Klasifikasi dikarenakan terhubung langsung secara one-to-many terhadap salah satu tabel pada entitas Komponen yaitu tabel *Product*. Hal tersebut dapat dilihat pada gambar 3.19



**Gambar 3.18** Relasi tabel *Product\_type* terhadap tabel *Product*



### 3.3.2 Identifikasi Hierarki (Identify Hierarchies)

Pada tahap ini dilakukan identifikasi hierarki sesuai pada tahapan sebelumnya. Identifikasi hierarki dilakukan untuk membentuk model dimensi / tabel dimensi dari *ER OLTP* yang ada [8].

#### 3.3.2.1 Hierarki Maksimal

*Hierarki maksimal* adalah sebuah kondisi entitas yang tidak dapat dilakukan ekspansi keatas ataupun kebawah dengan menyertakan entitas lain [8]. Pada penelitian ini, terdapat 6 *hierarki maksimal* yaitu

- *product\_type, product, price\_history*
- *product\_type, Product, Buy\_detail*
- *condition\_type, condition, product\_discount, product, price\_history*
- *condition\_type, condition, product\_discount, product, buy\_detail*
- *region, location, user, buy, buy\_detail*
- *buy\_methode, buy, buy\_detail*

Berdasarkan pada penjelasan tersebut, bisa dilakukan kategorisasi entitas. Tabel yang masuk pada hierarki *Entitas Minimal* adalah tabel *product\_type*, *condition\_type*, *region* dan *buy\_methode*. Sedangkan tabel yang masuk pada hierarki *Entitas Maksimal* adalah tabel *price\_history* dan *buy\_detail*. Tahap ini juga memiliki fungsi untuk melakukan identifikasi tabel mana yang berada posisi paling bawah dan paling atas sehingga dapat mempermudah dalam melakukan identifikasi tabel *sub-dimensi*, *dimensi* dan tabel *fakta*.

### 3.3.3 Perancangan Dimensional (Produce Dimensional Models)

Pada tahap ini dilakukan pembentukan *model dimensional* dari *ER OLTP* berdasarkan pada proses-proses yang dilakukan sebelumnya yang ada. Terdapat 2 operator yang dilakukan pada tahap ini yaitu operator *Collapse Hierarchy* (Peleburan Hierarki) dan *Aggregates Operator* (Operator Agregasi/perhitungan).

#### ***3.3.3.1 Collapse Hierarchy (Peleburan Hierarki)***

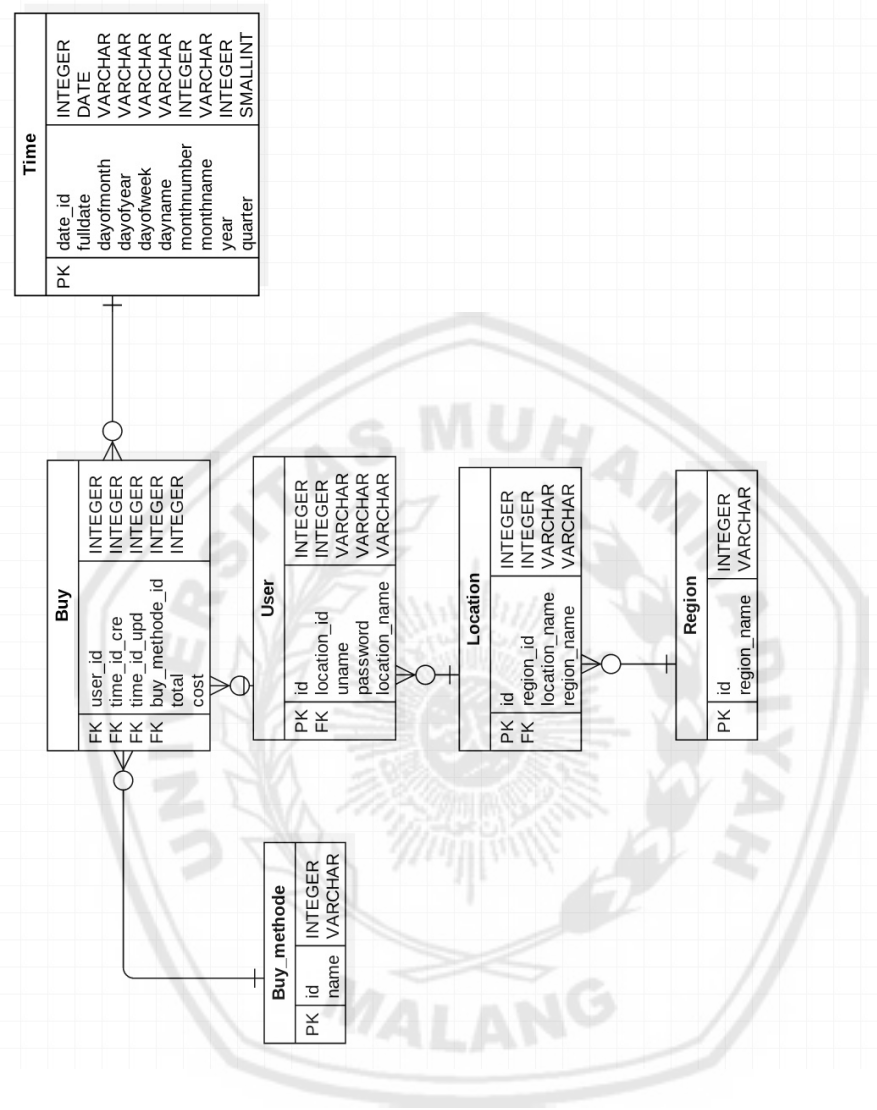
Pada penelitian ini tidak diperlukan peleburan hierarki dikarenakan skema yang digunakan adalah *snowflake* schema sehingga tabel pada entitas minimal tidak dilakukan tindakan apapun.

#### ***3.3.3.2 Aggregates Operator (Operasi Agregasi/Perhitungan)***

Pada operator *Agregasi* tidak dilakukan dikarenakan pada setiap kolom yang ada pada entitas transaksi sudah mewakili perhitungan yang dibutuhkan. Pada tahap ini dapat ditunjukkan hasil perancangan skema OLAP dari beberapa proses sebelumnya pada setiap transaksi yang ada dalam bentuk Snowflake Schema.

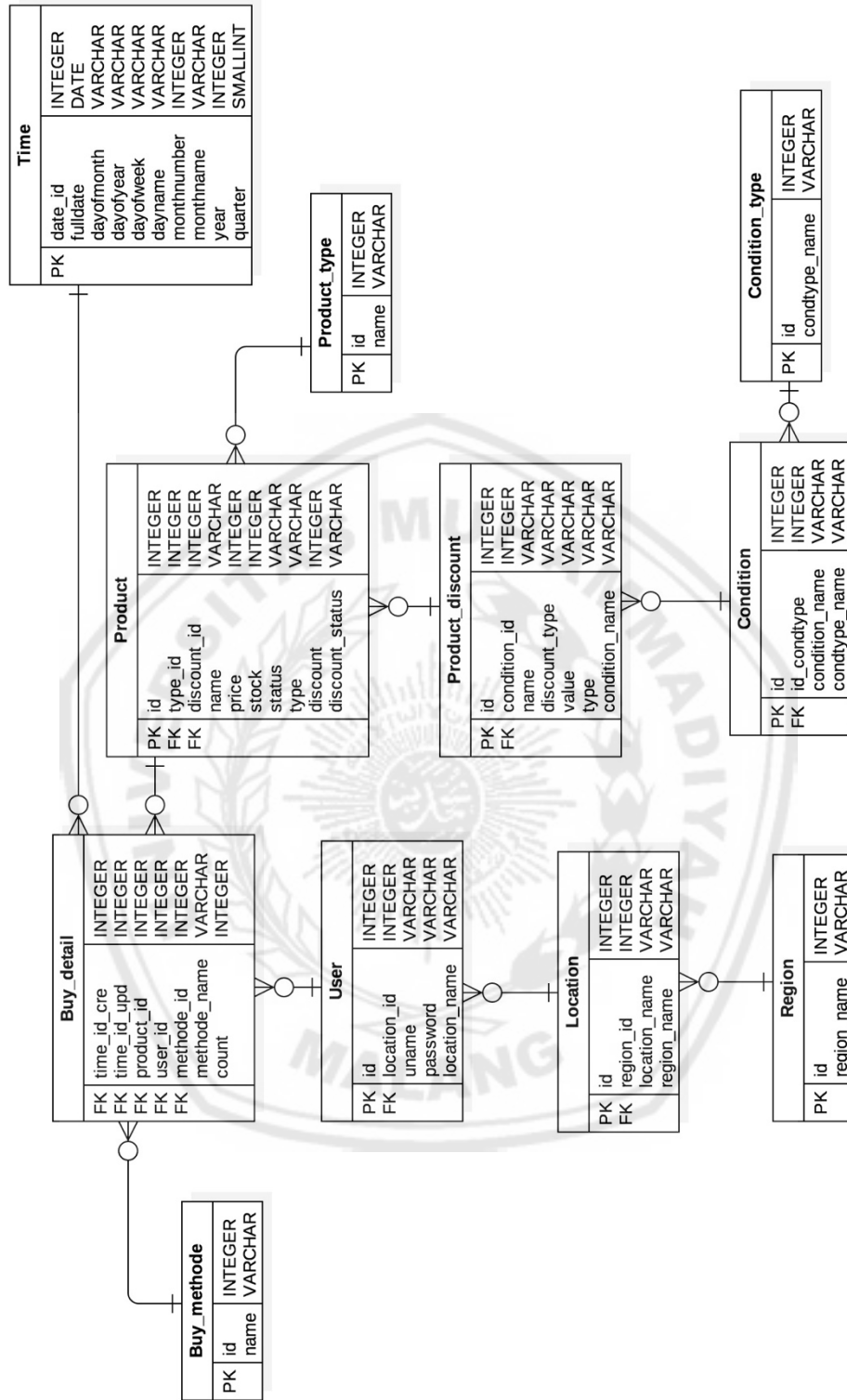


Snowflake Schema OLAP Buy



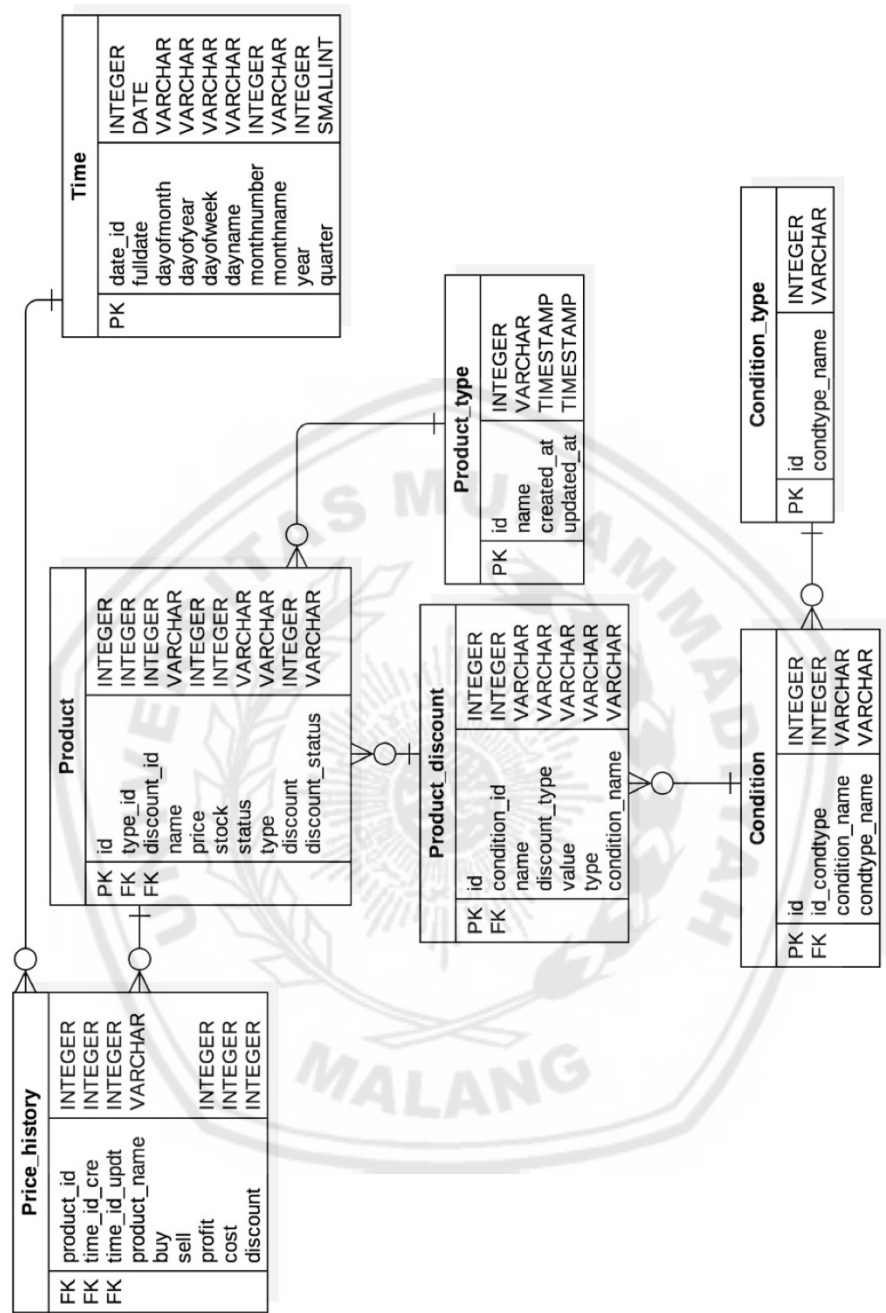
Gambar 3.19 Snowflake Schema OLAP Buy

Snowflake Schema OLAP Buy\_detail



Gambar 3.20 Snowflake Schema OLAP Buy\_detail

Snowflake Schema OLAP Price\_history



Gambar 3.21 Snowflake Schema OLAP Price\_history

### 3.3.4 Evaluasi dan Perbaikan (Evaluation and Refinement)

Tahap Evaluasi dan Perbaikan adalah tahap terakhir dari metode *ER Models to Dimensional Models*. Setelah dilakukan perancangan pada tahap-tahap sebelumnya, dilakukan sebuah evaluasi apakah rancangan yang dihasilkan sudah sesuai atau tidak. Rancangan *OLAP* yang sesuai menurut moody adalah tidak adanya pola non-hierarki pada entitas yang ada (moody). Pada rancangan *OLAP Virtual Shop* sudah sesuai dan bersifat hierarki pada setiap entitas yang ada sehingga tidak diperlu dilakukan tindakan pada tahap ini.

## 3.4 Fitur pada rancangan OLAP Virtual Shop

Rancangan *OLAP Virtual Shop* di bangun menggunakan paradigma Snowflake Schema. Perancangan menghasilkan tiga rancangan *OLAP* yaitu *OLAP Fakta Buy*, *OLAP Fakta Buy\_detail*, dan *OLAP Fakta Price\_history*. Pada masing-masing rancangan *OLAP* yang sudah dibentuk memiliki beberapa fitur yang akan dijelaskan pada bagian ini.

### 3.4.1 OLAP Fakta Buy

*OLAP Fakta Buy* merupakan rancangan *OLAP* yang digunakan sebagai data analisis yang berkaitan dengan data pembelian pada *Virtual Shop*. Pada rancangan ini terdiri dari 6 tabel yaitu tabel *Buy*, Tabel *User*, Tabel *Location*, Tabel *Region*, Tabel *Buy\_methode*, dan Tabel *Time*.

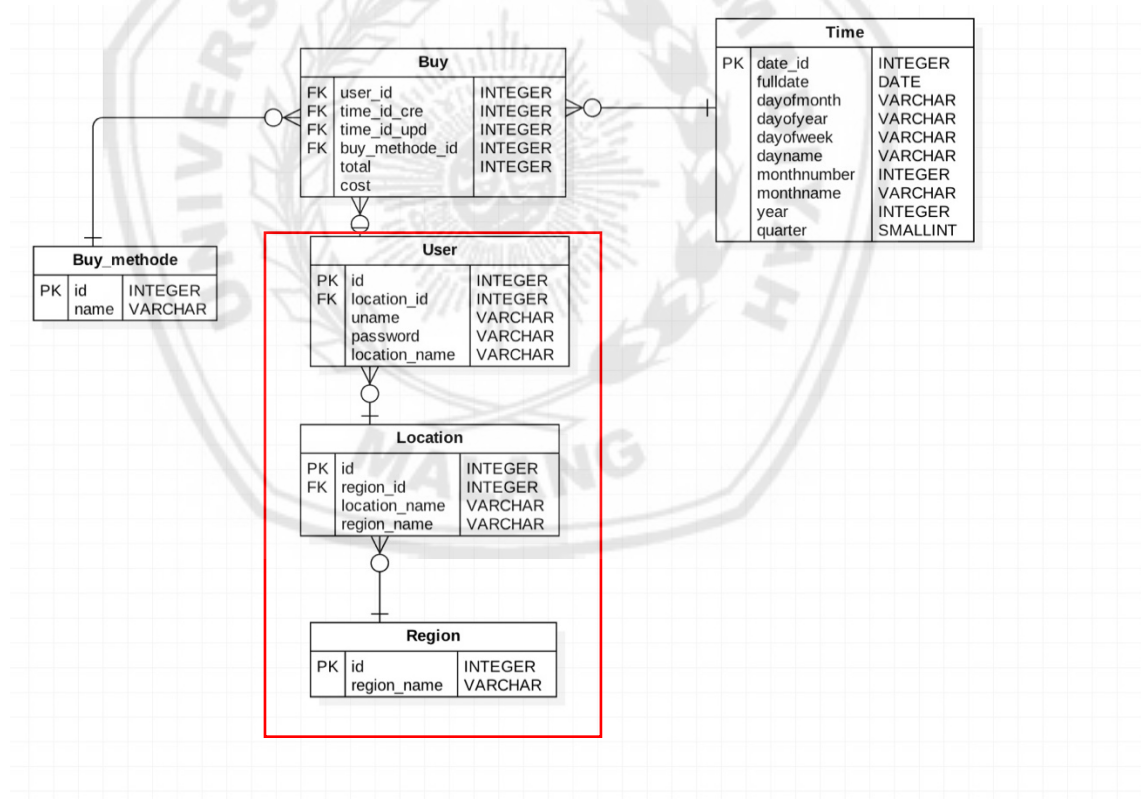
Fungsi-fungsi tabel yang ada dijelaskan melalui tabel 3.4.

**Tabel 3.4** Fungsi tabel pada *OLAP Fakta Buy*

| No. | NAMA TABEL | KETERANGAN  |
|-----|------------|---|
| 1.  | Buy        | Tabel digunakan untuk menyimpan data nominal pembelian yang dilakukan |
| 2.  | User       | Tabel digunakan untuk menyimpan data user yang melakukan pembelian    |

|    |             |   |
|----|-------------|---|
| 3. | Location    | Tabel digunakan untuk menyimpan data yang berkaitan dengan lokasi dalam pembelian   |
| 4. | Region      | Tabel digunakan untuk menyimpan data yang berkaitan dengan regional dalam pembelian |
| 5. | Buy_methode | Tabel digunakan untuk menyimpan data berkaitan dengan metode pembelian              |

Perancangan OLAP Fakta *Buy\_detail* menggunakan paradigma *snowflake schema* sehingga rancangan pada tabel dimensi dilakukan normalisasi secara hierarki dan membentuk tabel sub-dimensi agar tidak terjadi redundansi data. Tabel yang dilakukan normalisasi adalah pada tabel dimensi User yang dapat dilihat pada gambar 3.22.



**Gambar 3.22** Normalisasi tabel dimensi pada OLAP fakta Buy

Pada tabel *Buy\_methode* dan tabel *Time* tidak dilakukan normalisasi dikarenakan tabel tersebut sudah ternormalisasi.

### 3.4.2 OLAP Fakta Buy\_detail

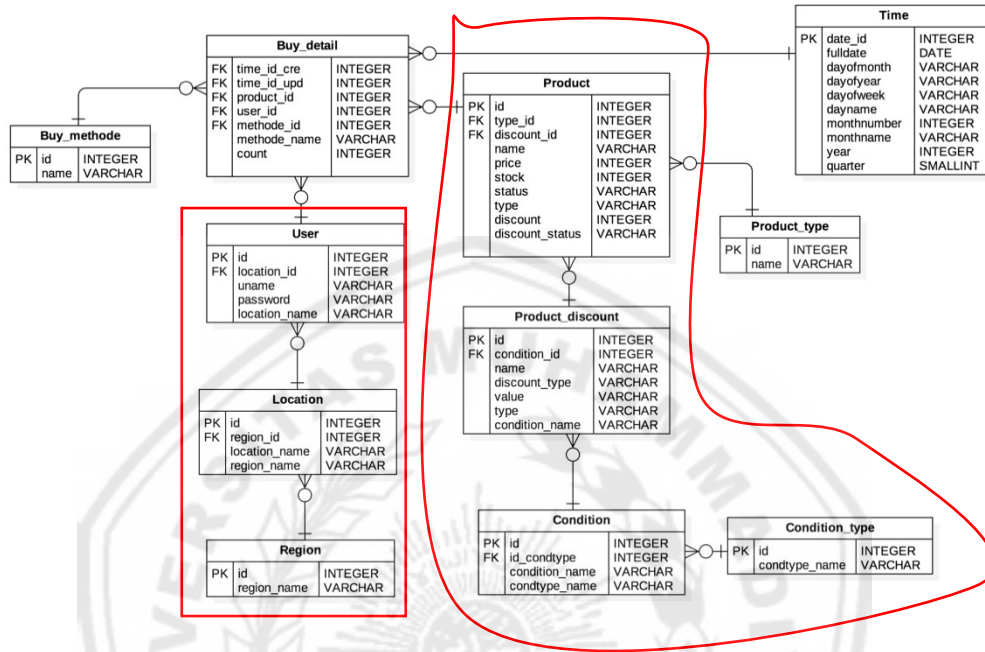
OLAP Fakta *Buy\_detail* merupakan rancangan OLAP yang digunakan untuk melakukan analisis terhadap data yang berkaitan dengan total barang yang terjual pada satu transaksi. Pada rancangan ini terdiri dari 11 tabel yaitu tabel *Buy\_detail*, tabel *User*, tabel *Location*, tabel *Region*, tabel *Buy\_methode*, tabel *Product*, tabel *Product\_discount*, tabel *Condition*, tabel *Condition\_type*, dan tabel *Product\_type*.

**Tabel 3.5** Tabel pada OLAP fakta Buy\_detail

| No. | NAMA TABEL       | KETERANGAN   |
|-----|------------------|--|
| 1.  | Buy_detail       | Tabel yang digunakan untuk menyimpan data detail dari pembelian pada sebuah transaksi.                             |
| 2.  | User             | Tabel yang digunakan untuk menyimpan data User dari kegiatan transaksi pembelian.                                  |
| 3.  | Location         | Tabel yang digunakan untuk menyimpan data Lokasi dari kegiatan transaksi yang berlangsung.                         |
| 4.  | Region           | Tabel yang digunakan untuk menyimpan data Regional lokasi dari kegiatan transaksi yang berlangsung.                |
| 5.  | Buy_methode      | Tabel yang digunakan untuk menyimpan data metode pembelian yang digunakan dari kegiatan transaksi yang berlangsung |
| 6.  | Product          | Tabel yang digunakan untuk menyimpan data produk yang yang dijual pada kegiatan transaksi                          |
| 7.  | Product_discount | Tabel yang digunakan untuk menyimpan data diskon pada produk yang ada pada kegiatan transaksi.                     |
| 8.  | Condition        | Tabel yang digunakan untuk menyimpan data kondisi diskon yang ada pada kegiatan transaksi.                         |
| 9.  | Condition_type   | Tabel yang digunakan untuk menyimpan data tipe kondisi diskon yang ada pada kegiatan transaksi.                    |
| 10. | Product_type     | Tabel yang digunakan untuk menyimpan tipe produk yang dijual   |



Perancangan OLAP fakta *Buy\_detail* menggunakan paradigma *snowflake schema* sehingga masing-masing tabel dimensi dilakukan normalisasi dan membentuk tabel sub dimensi. Tabel yang dilakukan normalisasi adalah tabel dimensi Product dan tabel dimensi User yang dapat dilihat pada gambar 3.23.



**Gambar 3.23** Normalisasi tabel dimensi pada OLAP *Buy\_detail*.

Pada tabel Time, tabel Buy\_methode, dan tabel Product\_type tidak dilakukan normalisasi dikarenakan sudah dalam bentuk normalisasi.

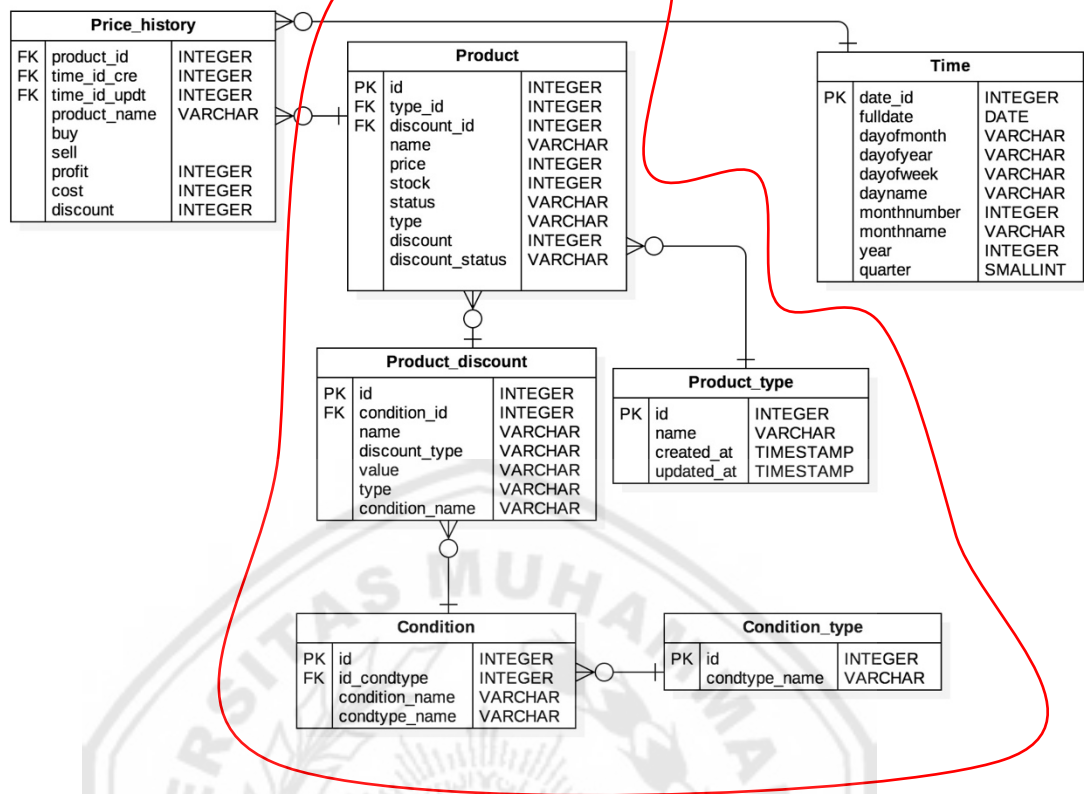
### 3.4.3 OLAP Fakta Price\_history

OLAP fakta Price\_history merupakan rancangan OLAP yang digunakan untuk melakukan analisis mengenai data harga yang ada pada sebuah produk yang dijual. Pada rancangan ini terdiri dari 7 tabel yaitu Tabel Product, tabel Price\_history, tabel Product\_discount, tabel Condition, tabel Condition\_type, tabel Product\_type, dan tabel Time.

**Tabel 3.6** Daftar tabel pada OLAP fakta Price\_history

| No. | NAMA TABEL       | KETERANGAN  |
|-----|------------------|---|
| 1.  | Price_history    | Tabel digunakan untuk menyimpan data detail harga pada produk yang dijual             |
| 2.  | Product          | Tabel digunakan untuk menyimpan data produk yang dijual.                              |
| 3.  | Product_discount | Tabel yang digunakan untuk menyimpan data diskon pada produk yang dijual              |
| 4.  | Condition        | Tabel yang digunakan untuk menyimpan data kondisi diskon pada produk yang dijual.     |
| 5.  | Condition_type   | Tabel yang digunakan untuk menyimpan data tipe kondisi diskon pada produk yang dijual |
| 6.  | Product_type     | Tabel yang digunakan untuk menyimpan tipe produk yang dijual.                         |
| 7.  | Time             |   |

Tabel dimensi pada OLAP ini yang dilakukan normalisasi adalah pada tabel dimensi Product, hal tersebut dapat dilihat pada gambar 3.24.



**Gambar 3.24** Normalisasi tabel dimensi pada *OLAP Price\_history*